

Patent

Docket No.: NVIDP021/P000174

**Petition to Make Special for Application Under MPEP §708.02(viii)****IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventor(s): John Erik Lindholm et al. Filing Date: 05/31/00 Serial No.: 08/586,249

Title: **SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR A PROGRAMMABLE VERTEX PROCESSING MODEL WITH INSTRUCTION SET**The Commissioner of Patents and Trademarks  
Washington, D.C. 20231Attention: Group Director, Group  
Sir:

(MPEP §1002.02(c))

**Petition to Make Special for Application**  
**(Under MPEP §708.02(viii))**

Applicant hereby petitions to make this application, which has not received any examination by the Examiner, special.

- ☒ All the claims in this case are directed to a single invention.
- ☒ If the Office determines that all the claims presented are not obviously directed to a single invention, Applicant will make an election without traverse as a prerequisite to the grant of special status. If claims 1-28 are found not to be examinable in this case with claim(s) 29-38, Applicant hereby elects claim(s) 29-38 for the prosecution of this case.

A search has been made by:

- ☐ the inventor ☒ agent
- ☐ professional searcher ☐ foreign patent office
- ☒ in the following:
- field of search: classes 345, subclasses ALL using the following search query:  
[opengl or (graphics and application and program and interface)]
- publications:
- foreign patents:
- search by corresponding foreign patent office or at the former International Patent Institute at The Hague, Netherlands.

- ☒ There is submitted herewith a copy of the references deemed most closely related to the subject matter encompassed by the claims.

☒ Form PTO-1449 is attached.

- ☒ There is submitted herewith a detailed discussion of the references which discussion particularly points out how the claimed subject matter is distinguishable over the references.

☒ A check in the amount of \$130.00 is enclosed.

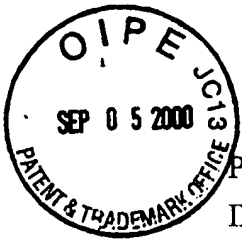
☒ At any time during the pendency of this application, please charge any fees required or credit any overpayments to Deposit Account 50-1351. A duplicate copy of this transmittal is enclosed.

Charge the Total Fees due to Deposit Account 50-1351. At any time during the pendency of this application, please charge any fees required or credit any overpayments to Deposit Account 50-1351. A duplicate copy of this transmittal is enclosed.

09/14/2000 SDUONG 00000125 09586249

01 FC:122

Date: 8/30/00 130.00 OP  
Kevin J. Zilka  
P.O. Box 721030  
San Jose, CA 95172-1030Kevin J. Zilka  
Reg. No. 41,429



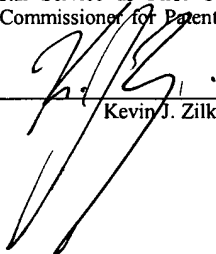
PATENT  
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the application of: ) Docket: NVIDP021/P000174  
)  
Lindholm et al. )  
) Group Art Unit: Unknown  
Serial No.: <sup>09</sup> ~~08~~586,249 )  
) Date: August 31, 2000  
Filed: May 31, 2000 )  
)  
For: SYSTEM, METHOD AND )  
ARTICLE OF MANUFACTURE )  
FOR A PROGRAMMABLE )  
VERTEX PROCESSING MODEL )  
WITH INSTRUCTION SET )

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231 on August 31, 2000.

Signed: \_\_\_\_\_

  
Kevin J. Zilka

Assistant Commissioner for Patents  
and Trademarks  
Washington, DC 20231

Dear Sir:

In association with a petition to make special filed herewith, following is a detailed discussion of the references of record which particularly points out how the claimed subject matter is distinguishable over the references.

**OpenGL Extension Documentation**

NVIDP021/P000174

<http://www.opengl.org/Documentation/Extensions.html?extensions>

<http://oss.sgi.com/projects/ogl-sample/registry/doc/rules.html>

The present documentation relates to OpenGL which is the computer industry's standard graphics application program interface (API) for defining 2-D and 3-D graphic images. With OpenGL<sup>®</sup>, an application can create the same effects in any operating system using any OpenGL<sup>®</sup>-adhering graphics adapter. OpenGL<sup>®</sup> specifies a set of commands or immediately executed functions. Each command directs a drawing action or causes special effects. In particular, the present documentation discloses that OpenGL<sup>®</sup> offers extensions, as well as extensions to related APIs like GLU, GLX, and WGL, have been defined by vendors and groups of vendors. The OpenGL<sup>®</sup> Extension Registry is maintained by SILICON GRAPHICS (SGI) and contains specifications for all known extensions, written as modifications to the appropriate specification documents. The registry also defines naming conventions, guidelines for creating new extensions and writing suitable extension specifications, and other related documentation.

The OpenGL<sup>®</sup> extensions described in the above documentation fail, however, to provide the ability to operate in a programmable mode during which programmable operations are performed. While the OpenGL<sup>®</sup> extensions do teach the ability to program one's own extensions, they do not facilitate such programming in the manner unique to the claimed invention. In particular, the OpenGL<sup>®</sup> extensions documentation fail to allow the operations to be programmed by a user "utilizing instructions from a predetermined instruction set which is different from an instruction set of a standard graphics application program interface," as claimed by applicant. In contrast, the OpenGL<sup>®</sup> extensions simply utilize the instructions associated with the OpenGL<sup>®</sup> graphics API in order to afford modified OpenGL<sup>®</sup> commands. By providing a different instruction set, the claimed invention is capable of providing "increased flexibility in programming the processing in the computer graphics pipeline" while "preserv[ing] hardware optimization afforded by the standard graphics application program interface," as claimed. Simply nowhere in the prior art is there disclosed, taught or suggested such a technique for providing a unique combination of features which affords the foregoing claimed benefits.

**U.S. Patent Number 6,046,747 to Saunders et al.**

**GRAPHICS APPLICATION PROGRAMMING INTERFACE AVOIDING  
REPETITIVE TRANSFER OF TEXTURE MAPPING DATA**

The present patent discloses a computer graphics system that includes a host computer, having a graphics application program and an application programming interface, and graphics hardware. The application programming interface creates a display list for receiving and storing user-specified commands and stores each user-specified texture download command in a texture image list. When a texture processing command is received and the corresponding texture map is optimizable, the texture image list is stored in a texture object, the texture object is marked as not downloaded and a bind texture call is inserted in the display list. The bind texture call contains a reference to the texture object. When the display list is executed, each bind texture call in the display list is processed by: (a) downloading the texture map referenced by the bind texture call to the graphics hardware when the referenced texture object is marked as not downloaded, and marking the referenced texture object as downloaded, and (b) binding texture information in the texture object to a current texture state, wherein the texture map is downloaded to the graphics hardware only once for a given texture state of the application programming interface.

The present patent fails, however, to disclose the steps of “determining whether the graphics pipeline is operating in a programmable mode,” “performing programmable operations on the graphics data in order to generate output if it is determined that the graphics pipeline is operating in the programmable mode” and “performing operations on the graphics data in order to generate output in accordance with the standard graphics application program interface if it is determined that the graphics pipeline is not operating in the programmable mode,” as claimed by applicant. In contrast, Saunders merely optimizes the transfer of texture mapping data utilizing a standard graphics application program interface such as OpenGL<sup>®</sup>, and does not address performing programmable operations on the graphics data.

**U.S. Patent Number 5,966,532 to McDonald et al.**

**GRAPHICAL CODE GENERATION WIZARD FOR AUTOMATICALLY  
CREATING GRAPHICAL PROGRAMS**

The present patent discloses a computer-implemented system and method for automatically generating graphical code in a graphical programming system. The computer memory stores a plurality of graphical code templates. The graphical programming system executing on the computer system also includes a plurality of front panel objects or controls which represent the user interface. One or more associated graphical code portions or templates can be associated with certain of the controls. According to the present invention, the user first selects a control and then preferably initiates the graphical code generation wizard for the control. When the graphical code generation wizard is invoked, the wizard displays on the screen a configuration panel or dialog, prompting the user to configure the control or object. The user then selects parameter values to configure certain aspects of the graphical code being created. The graphical code generation wizard selects a graphical code template in response to the control and configures the graphical code template with the parameter values. The graphical code generation wizard then creates an association between the control and the configured graphical code. The user can edit wizard created code either using the graphical code generation wizard or by unlocking the association between the control and the code and making the changes directly in the block diagram.

The present patent fails, however, to disclose a dual mode of operation, where it is determined “whether the graphics pipeline is operating in a programmable mode,” “programmable operations [are performed] on the graphics data in order to generate output if it is determined that the graphics pipeline is operating in the programmable mode” and “operations [are performed] on the graphics data in order to generate output in accordance with the standard graphics application program interface if it is determined that the graphics pipeline is not operating in the programmable mode.” On the other hand, McDonald merely teaches a mechanism for generating programmable operations,

not implementing a system that allows the use of programmable operations in addition to operations associated with a standard graphics application program interface.

**U.S. Patent Number 5,798,762 to Sfarti et al.**

**CONTROLLING A REAL-TIME RENDERING ENGINE USING A LIST-BASED  
CONTROL MECHANISM**

The present patent discloses a graphics system that has a multi-port memory subsystem storing image-rendering control lists and that further has a real-time programmably-configurable render-engine (also referred to herein as the 'triangle-engine' or 'TE') coupled to the memory subsystem and responsive to the stored image-rendering control lists as well as to supervisory immediate control commands issued by a supervisory CPU or the like. A triangle-engine in accordance with the invention reduces the load on system memory and on system CPU's by periodically fetching compact render-control lists and source image data from the memory subsystem on a list basis and by processing the fetched information without requiring continuous real-time intervention by system CPU's. This frees the system CPU (or CPU's) for managing other time-critical tasks of the graphics system.

The present patent discloses a system for offloading a CPU by pulling commands from memory which, in turn, control a programmably-configurable render-engine. In contrast, the invention claimed by applicant requires "performing programmable operations [that] are programmable by a user utilizing instructions from a predetermined instruction set which is different from an instruction set of a standard graphics application program interface in order to provide increased flexibility in programming the processing in the computer graphics pipeline."

**U.S. Patent Number 5,798,770 to Baldwin et al.**

**GRAPHICS RENDERING SYSTEM WITH RECONFIGURABLE PIPELINE  
SEQUENCE**

The present patent discloses a pipelined graphics processor in which the sequence can be dynamically reconfigured (e.g. between primitives) in a rendering sequence. The pipeline sequence can be configured for compliance with specifications such as OpenGL<sup>®</sup>, but may also be optimized by reconfiguring the pipeline sequence to eliminate unnecessary processing. In a preferred embodiment, pixel elimination sequences such as depth and stencil tests are performed before texturing calculations are performed, so that unneeded pixel data is discarded before said texturing calculations are performed.

The present patent fails, however, to disclose "performing programmable operations .... [that] are programmable by a user utilizing instructions from a predetermined instruction set which is different from an instruction set of a standard graphics application program interface in order to provide increased flexibility in programming the processing in the computer graphics pipeline," as claimed by applicant. Instead, the instant patent teaches configuring the rendering sequence in accordance with only a standard graphics application program, namely OpenGL<sup>®</sup>, not an instruction set which is different from an instruction set of a standard graphics application program interface, as claimed by applicant.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. If any fees are due in connection with the filing of this paper, then the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NVIDP021). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,

  
Kevin J. Zilka

Registration No. 41,429

P.O. Box 721030

San Jose, CA 95172

Telephone: (408) 505-5100

**NVIDP021/P000174**